Journal of Theoretical and Applied Information Technology
15th August 2023. Vol.101. No 15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ANALYSIS OF QOS ROUTING IN MOBILE AD HOC NETWORKS USING ANTHOCNET, SHUFFLED FROG LEAP, FIREFLY AND LION OPTIMIZATION ALGORITHMS

K.SUMATHI[1], DR.J.K.KANIMOZHI[2]

[1]Research Scholar, Periyar University, Salem, Tamilnadu, India.
[2]Associate Professor, PG & Research Department of Computer Science,
Vivekanandha College of Arts and Sciences for Women (Autonomous) ,Tiruchengode,Tamilnadu,India.

E-mail: sumathiksmsc2002@gmail.com, drjkkanimozhi@gmail.com

ABSTRACT

A Mobile Ad Hoc Network (MANET) is a network of wireless nodes. Base stations, infrastructure, and infrastructure are not required for a mobile ad hoc network. MANET is a network that is only available temporarily. MANET technology is used in military and search and rescue operations. Quality of Service (QoS) is a key term for the overall optimization of network resources. Mobile Adhoc networks are well-known for their self-organization and autonomy. QoS-based routing over MANET necessitates an adaptive and fast path search solution. Optimization algorithm-based techniques, such as ant colony optimization (ACO) algorithms, shuffled frog leaping algorithm, Lion Optimization Algorithm (LOA), and Firefly Algorithm (FF), have proven to be effective in developing routing algorithms for mobile ad hoc networks. Then, an optimal path is computed using hybrid ACO based routing, which is an efficient routing scheme based on foraging ant behaviour. The proposed hybrid algorithm outperforms the Anthocnet and shuffled frog leap, firefly, and lion optimization algorithms in simulation (LOA). The proposed SFLAO algorithm also outperforms the standalone Anthocnet and LOA, Shuffled frog leap, and Firefly algorithms in terms of QoS performance metrics.

Keywords: *Mobile Ad Hoc Network, Routing Protocol, Quality Of Service, Lion Optimization Algorithm, Ant Colony Optimization, Shuffled Frog Leaping Algorithm*

1. INTRODUCTION

The nodes in a mobile ad hoc network (MANET) collaborate in a distributed manner to enable routing between them [1]. Because of the lack of centralised control, routing becomes a central issue and a significant challenge as the network topology changes on a regular basis. It is a network of mobile nodes that are dynamically and arbitrarily located in such a way that node interconnections can change on a regular basis. Each node has the capability of acting as a receiver, transmitter, or router. The main issue with an ad-hoc network is the mobility of the nodes, which causes rapid variations in their availability. At one point, the node is in range, but at another, it is out of range. A mobile ad hoc network (MANET) is a decentralised network of mobile nodes that exchange information in the short term via wireless transmission. The network topology is unstructured, and nodes can join or leave at any time. A node with a transmission range can communicate with other nodes. Because there is no background network for centralised network control, network control and management are distributed. When transmitting data packets from a source to a destination, the packets should be routed via one or more intermediate nodes.

As the internet grows in popularity, so does the demand for real-time and quality of service (QoS) in the network. A QoS routing strategy's role is to compute paths that are suitable for various types of traffic generated by various applications while maximising network resource utilisation. However, the problem of finding multiconstrained paths has a high computational complexity, necessitating the use of algorithms that address this difficulty. The primary goals of QoS routing are,

- • Determine a path from source to destination that meets the needs of the user.
- • To optimise network resource usage and
  • To degrade network performance when

undesirable things such as congestion and path breaks appear in the network [2].

In order to maintain network performance, routing must be carefully considered as one of the major factors. On demand routing algorithms, as opposed to table-driven algorithms, do not form route information among nodes. Routes are only established when they are absolutely necessary. Routes are created only when they are required, that is, when any of the nodes wishes to send a packet. As a result, routing overload is lower than that of table-driven algorithms. As a result of each node not maintaining up-to-date route information, the packet delivery fraction is low. Routing algorithms' primary goal is to define a path from source to destination that maximises network performance while minimising costs. Furthermore, routing focuses on diagnosing, enrolling, and disseminating the paths from the source node to the destination node. However, it appears to be a difficult task in MANETs because the participating nodes in this network move in an arbitrary manner without constant speed, resulting in connectivity issues that may split the connection among the hosts. The routing protocols in MANET's are categorized in three classes: proactive, reactive and hybrid [3][4][5][6].

The nature of nodes and their mobility are critical factors in MANET performance. Because MANET devices have limited transmission range, they cannot always communicate directly with the destination device. As a result, communication is routed via intermediate nodes. Because of the mobility of nodes in MANETs, nodes frequently enter and exit the transmission range, interfering with MANET routing. As a result, in order to support the routing function, nodes frequently exchange data in order to be aware of the network's status. MANETs are extremely powerful and are widely used in a variety of real-world scenarios such as battle scenarios, rescue operations, and vehicular networks, where traditional network infrastructure is difficult or impossible to set up. The topology of the MANET changes constantly due to node mobility. Taking these considerations into account, the Routing Algorithm is subject to some additional requirements. A MANET routing algorithm should not only be capable of finding the shortest path between the source and destination, but it should also be adaptive to changing node states, network load conditions, and environmental changes.

The goal is to achieve optimal QoS in MANETs. To improve QoS, several techniques have been proposed. The basic function of QoS routing is to find a network path that satisfies the constraints that have been specified. QoS is a term used to measure the goals of both the service provider and the service requester; the QoS composition process will select the task based on metrics and aggregate the services that have obtained the most value by meeting the needs of the system. The goal of QoS provisioning is to achieve more deterministic network behaviour, so that network information is delivered correctly and network resources are better utilised. However, providing QoS solutions and maintaining end-to-end QoS while allowing for user mobility remains a significant challenge. When compared to wireless networks, implementing Quality of Service in wired networks is simpler. We can increase the number of resources in demand in a wired network, and application-specific QoS requirements can be reserved. In a wired network, traffic can also be differentiated based on QoS and other factors. There are numerous challenges in implementing the aforementioned QoS requirements in wireless networks.

Few of the drawbacks are listed as follows:

- The MANETs' bandwidth is very low due to mobility and resource constraints.
- Because MANETs have a dynamic topology, nodes frequently disconnect.
- Due to limited resources and computational capacities, complex data processing may be impossible.Because MANET nodes have limited computational capabilities, it is difficult to maintain the status of updated network information.
- Most QoS mechanisms require packet signalling. This results in a competition for network resources as well as data packets.
- The lack of observation and supervision of admission control in MANETs may reduce the effectiveness of QoS..

In recent years, lightweight QoS mechanisms that are easy to compute have been proposed for MANETs[7]. Source routing, distributed routing, and hierarchical routing are the three types of routing strategies. QoS-based routing becomes difficult in MANETs because nodes must keep track of link status. Furthermore, because MANETs are dynamic, it is difficult to maintain precise link state information [8]. Finally, the reserved resource may not be guaranteed due to mobility-related path breakage or mobile hosts' power depletion. QoS routing should find a viable new route to the service as soon as possible.

2. OPTIMIZATION ALGORITHMS

Mobile Ad hoc networking with an optimization algorithm that finds the path only when it is requested, that is, when a node has data to send to other nodes.Because the problem of finding paths has a high computational complexity, optimization algorithms such as those discussed below are required.

2.1. Ant Colony Optimization Algorithm

Ant Colony Optimization with MANET (AntHocNet) seeks to create an algorithm that can operate efficiently in MANETs while retaining the properties that make ACO routing algorithms so appealing. While most previous wired network algorithms used a proactive approach, generating ant-like agents on a regular basis for all possible destinations, AntHocNet uses a hybrid approach. Ants are produced using both proactive and reactive strategies. It is reactive in the sense that it only gathers routing information about destinations that are involved in communication sessions. It really is proactive in the sense that it attempts to maintain, improve, and extend routes while the communication session is in progress. By concentrating efforts on ongoing sessions, this hybrid architecture improves efficiency [13[14]. Another factor that contributes to efficiency is the organisation of the proactive route maintenance and improvement process, which combines ant-based path sampling with other forms of information gathering. Pheromone tables are used to store routing information in AntHocNet. These tables are used to perform stochastic forwarding of ant and data packets. When a link fails, specific reactive mechanisms, such as local route repair and the use of warning messages, are used to deal with it.

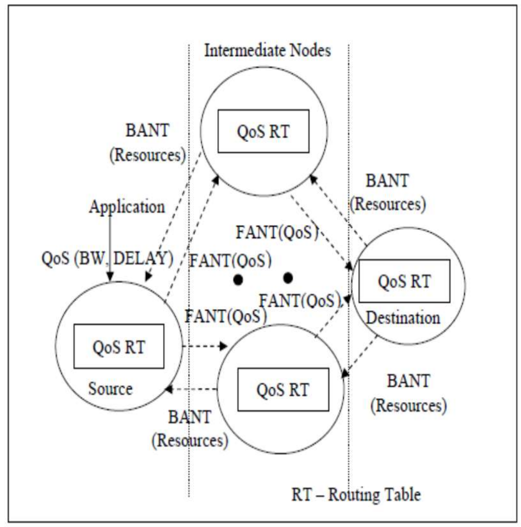


Figure 1: AntHoc –Quality of Services: Architecture Diagram

The algorithm employs two kinds of ants: forward ants (FANT) and backward ants (BANT) (BANT). Ants build a solution incrementally by moving to feasible neighbour states from the start state. A probabilistic decision rule is used to select the next state at each state. The goal of ACO algorithms is the local stigmergetic actions resulting in emergent global behaviour. Pheromone trail evaporation prevents the solution from rapidly settling into a suboptimal state due to an uncontrolled autocatalytic effect. Broadcasting HELLO messages initialises the routing table and performs neighbour discovery. When the first FANT is sent to a specific destination, the routing tables are initially bootstrapped. (Figure. 1). The routing table entries are seeded with equal probabilities 1 / N for each neighbour to be the next hop for the respective destination. The probabilities are adjusted by BANT sent from the destination. The probability *Pi.j* of choosing a neighbour *vj* ∈ *Ni* at any node vi is given by:

$$P_{i,j} = \frac{p_{QOs,ij}}{\sum_{v_j \in N_i} p_{QOS,ij}} \, if \;\; v_j \in N_i$$

$$= 0 \;\; i \;\; fv_j \notin N_i$$

2.2. Lion Optimization Algorithm

The LOA algorithm chooses the best route from the source to the destination from among all possible routes. The LOA is a bio-inspired algorithm that mimics lion behaviours. Lions are social animals that live in prides. A pride of lions includes both resident and nomadic lions [9],[10]. Each pride

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

consists of four to five lions, including cubs and adult lions. When the young lions reach adulthood, they are dislodged from the pride and become nomadic lions. Because nomadic lions are mostly young, they attack the adult lions in the pride in order to gain entry. The LOA algorithm creates an initial lion population that includes both pride and nomadic lions. In addition to the pride's control region, the ratio of nomadic and resident lions is set. The gender ratio is also fixed, as lionesses prioritise hunting. This bio-inspired algorithm finds the best solution by arranging the nomadic lions according to their fitness value. The lions with the highest fitness values are given a place in the pride, while the lions with the lowest fitness values are discarded. This operation will be repeated until no better solution can be found.

Algorithm:

Step 1: Establish route in cross layer network
Step 2: Find the node in network is trustworthy.
Step 3: Frequent transmission
Step 4: If node= =anomaly
Step 5: Communication is obstruct
Step6: Overload occurred
Step 7: Else
Step 8: If node! =anomaly
Step 9: Communication can be performed
Step 10: Improves network lifetime and Reduce delay.
Step 11: End
Step 12: End process

Initialization process : The input solution is first generated at random across the search space. The number of partitions is used as the input solution here. Set up the random partitions Li ¼ ½p1; p2; ...; pN,where N is the total number of partitions chosen from the search space..

***Fitness Computations*:** An objective function evaluates the fitness values of lions in the sorted and saved matrix

$$f(Lion) = f(x1,x2,\dots, xNvar) \quad (9)$$

Fitness values are based on energy and delay or the effectiveness with which the forwarder node forwards emergency messages.

Hunting Operation: Hunters are classified into three types. The centre hunter has the highest fitness value, while the other two form the left and right sides. Hunters are chosen at random to attack dummy preys that escape when the hunter's fitness improves and he is relocated to a new place.

Defense: New mature males in the pride will fight other males. The weakest males will abandon their pride and become nomads. By combining new mature males and old males, this behaviour can be simulated. The males are then sorted based on their fitness levels. The pride's weakest males are expelled and become nomads, while the remaining males become resident males. This strategy helps our proposed algorithm retain powerful male lions as solutions in LOA. Nomad males attack prides at random in an attempt to take over a pride by fighting the pride's male lions. The weak male lion will be driven out of the pride and become a nomad if the nomad lion is strong enough.

Movements towards Safety: Only a few female lions hunt for prey, while the rest stay in safe territory. The best positions for each territory are calculated and saved. A high victory count indicates that the lions have deviated from the optimal point. Lower values indicate that lions are roaming for improvement, and thus competition evaluation indicates success.

Migration: Influenced with how lions migrate and change their lifestyles in nature, resident females of one pride become nomads in another, and so on, it broadens the diversity of the target pride by virtue of its membership in the first. The lion's migration and switch lifestyle, on the other hand, create a bridge for information exchange. The number of female lions in a pride is calculated as a percentage of the total population of the pride. Some female lions are chosen at random in a pride and become nomads for migration. The size of migrated female lions in each pride is equal to the surplus female lions plus %I of the maximum number of female lions. This procedure facilitates information sharing among prides and preserves population diversity.

Termination criteria: This algorithm ends the process if a large number of iterations with the best fitness are completed. The partition with the highest fitness is chosen as the best partition in which to broadcast the message first.

2.3. Firefly Algorithm

This firefly algorithm was inspired by the swarm behaviour of fireflies. Fireflies are generally thought to exist in groups and exhibit swarm-like behaviour. The blinking light in fireflies is an attractive trait that they use to attract mates and defend themselves from other predators. The swarm of fireflies usually moves in the direction of the brightest one. All of the other fireflies with lower light intensities migrate to the ones with higher light intensities. As the distance between the fireflies increases, so does the intensity of the light. The firefly algorithm is based on the concept of the enlightening lights of the firefly. The intensity of the light aids a firefly's group in shifting to more intense and appealing positions that are plotted to produce the best resolution over the seeking location. This mechanism normalises a few firefly characteristics, as illustrated below [11,12]:

Every firefly is attracted to a different irrespective of their sex.

- The firefly's brightness is openly comparable to its attractiveness; the firefly with more brightness attracts the one with less brightness.
- A firefly shifts arbitrarily when it is unable to find a brighter and closer firefly.

Light intensity and attractiveness are two important factors in the Firefly algorithm. The degree of attractiveness determines the distance that a firefly has moved, while light intensity reflects the pros and cons of the firefly's location and determines its direction of movement. To achieve the goal of optimization, light intensity and attractiveness are constantly updated. The firefly's optimization mechanism is mathematically described as follows.

Light intensity and attractiveness: The firefly algorithm's execution is dependent on the disparity in light intensity and the conceptualization of attractiveness; light intensity represents a firefly's brightness and decreases with distance from its source. Furthermore, as a result of light absorption by air, attractiveness varies with the degree of absorption (Equation 4).

$$I = I(4)\, o \exp^{-\gamma r^2} \text{ ---- (1)}$$

where Io denotes the initial intensity of light and is the absorption coefficient The higher the light intensity, the better the order of requests in the request queue. A firefly's attractiveness is limited by the brightness associated with the ciphered fitness function.

A firefly's fluorescence intensity is defined as,

$$I(r) = I_o e^{-\gamma r^2} \quad \text{--- (2)}$$

Where I 0 represents the initial fluorescence intensity at r = 0, and is the fluorescence absorption coefficient. Because firefly attractiveness is proportional to the fluorescence intensity seen by neighbouring fireflies, firefly attractiveness is proportional to I(r), which can be written as

$$B(r) = B_0 e^{-\gamma r^2} \quad \text{---- (3)}$$

$B(r)$ is the fireflies attractiveness at a distance (r), and when $r = 0$, then the attractiveness is $B0$. i and j are supposed to be two fireflies with $Xi(xi, yi)$ and $Xj(xj, yj)$ as positions. Among two fireflies, the distance rij is computed based on the -Euclidean distance which is given in Equation 4.

$$r_{ij} = \| x_i - x_j \| = \sqrt{(x_i - x_j) - (y_i - y_j)^2} \quad \text{---- (4)}$$

Movement of Firefly:

If firefly i is drawn to the brighter firefly j, its movement is defined by the equation [13, 14].
Thus, Equation 5 determines the new position (Xi) and movement of firefly i' towards j.

$$x_i = x_i + B_0 e^{-\gamma r^2_{ij}} \ (X_j - X_j) + \alpha \in_i \quad \text{------- (5)}$$

In Equation 5, i is the random variables vector and the R and optimization factor (α) [0, 1].

Algorithm:

Initial population of n fireflies x(t)=[x 1,x 2. . . . .x d]
Divide population into 3 clusters
Generate objective function fn(x),
Generate objective function g(x)
Light intensity Ii determined by g(x)
Define light absorption coefficient
For k=1:n
While (t<MaxGeneration),
If g(k) = inf
Mutate k to new k using b0=0, and b0=1,
for invalid and valid nodes for each x (i) respectively.
Each x (i)moves probabilistically towards the valid nodes closest to it.
If g(new k) < g(best k)
best k = new k
End If
End If
End While
End for k
Report best_k as final route
Post process results and visualization

2.4. Shuffled Frog Leap Algorithm (SFLA)

Nature is a wonderful, inspiring source of phenomena with a dynamic, abundant display for solving highly complex and difficult problems in various disciplines. Nature-inspired meta-heuristic algorithms are intelligent technologies that mimic nature in order to solve optimization problems and develop new methods in computer calculations. SFLA is one of the nature-inspired algorithms. The population of SFLA is made up of a group of frogs looking for food in a pool. Food hunting is divided into two stages: the intra-group relationship of frogs that belong to different memeplexes of general evolution[15][16]. At the moment, there is no unified understanding of how SFLA effectively applies to the discrete optimization problem. The common method is to redefine frog particle encoding in accordance with MANET characteristics, and to introduce "the conversion gene" and "conversion sequence" concepts in the traditional local search mechanism that correspond with the frog particle encoding mode to improve the algorithm search breadth and speed.

Initial population: Assume in the SFLA that the initial population consists of F randomly generated frogs Xi. Then, using a known method, evaluate the fitness fit(i) of the ith frog and sort the frogs in descending order of fitness values to divide into m memeplexes Y 1, Y 2,..., Y m. Assume n frogs in each memeplex for m memeplexes. It is self-evident that F = m n. The memeplex Y k can be built by,

$$Y^k = \{X_i^k \mid X_i^k = X_{K+M}(i-1), i = 1,2,...n\}$$
$$k = 1,2,.....,m$$
(1)

The first frog visits the firstmemeplex, the second frog visits the second memeplex, the mth frog visits the mth memeplex, and the (m + 1)th frog returns to the firstmemeplex. The next step is to run local evaluation processes in parallel in each submemeplex to repeatedly update the worst frog's position. The memeplex's worst frog's new position is updated according to the following rule:

$$D = rand.(X_{sbest} - X_{worst})$$
$$X_{worst}^{k+1} = X_{worst}^{k} + D, \quad D_{\min} \le D \le D_{MAX}$$
(2)

where Xsbest and Xworst represent the best and worst frog positions in the submemeplex; D denotes the change in frog step size; Dmin and Dmax denote the minimum and maximum allowable change in frog position, respectively; rand is a random number between 0 and 1; k is the submemeplex iteration number. If Xk+1 worst outperforms the original worst performance Xkw orst, Xkworst should be replaced with Xk+1 worst. Otherwise, instead of Xsbest, the global best frog Xgbest is used to carry out the above updating strategy. The following modifications can be made to the operation process

$$D = rand.(X_{Gbest} - X_{worst})$$
$$X_{worst}^{k+1} = X_{worst}^{k} + D, \quad D_{\min} \le D \le D_{MAX}$$
(3)

If in this situation there is still no improvement, the Xworst is replaced with a random feasible solution. Within each memeplex, the process is repeated for a predetermined number of memetic evolution steps. The frogs are then shuffled for global information exchange. The local exploration and shuffling processes alternate until the defined convergence criterion is met.

Algorithm:

Begin Generate random population of P solutions (frogs);
For each individual $i \in p$ calculate fitness (i);
Sort the population P in descending order of their fitness; Divide P into o m memeplexes;
For each memeplex determine the best and worst frogs
improve the worst frog position using equations(1)or(2);
repeat for a specific number of iterations;
End;
Combine the evolved memeplexes;
Sort the population P in descending order of their fitness;
Check if termination =true;
End

Where rand () is a random number between 0 and 1, and Dmax is the maximum allowable change in frog position. If the same process can produce a better solution, it will now replace the frog found to be the worst, and if not, the calculations in (1) and (2) will be repeated in relation to a frog that is the global best (the Xg replaces the Xb). If no improvement is seen, there is another new solution that has been generated at random for replacing a frog. These calculations will now be repeated for a set number of iterations.

$$change\ in\ \ frog position\ (D_i) = rand() \times (X_b - X_w)$$
$$New\ position\ X_w = current\ \ position\ X_w + D_i$$
$$D\max \ge D_i \ge -D_{\max}$$
(2)

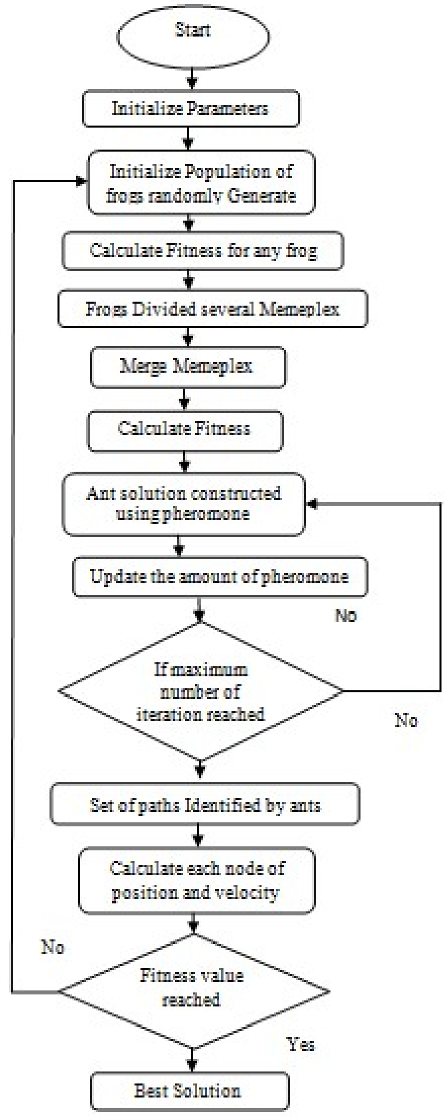
According to this, the SFLA's primary parameters were: the actual number of frogs P, the number of generations for each such memeplex even before a

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

shuffling occurs, the number of shuffling iterations, the maximum step size, and finally the number of memeplexes that are considered.

III. PROPOSED METHODOLOGY

The primary function of MANET in QOS routing data to any point in the network. The obvious solution is to route to the destination using the fewest steps possible. This technique reduces routing time significantly. In order to achieve better performance, one of the primary sources of energy consumption in MANET is communication. As a result, in order to extend network lifetime, Throughput, Routing Overhead, End to End Delay, and Packet Delivery Ratio must be optimised, which selects the optimal path. In order to improve the optimal performance of SFLAO, when a node receives a request, it first looks up a new population initialization scheme and then all accessible routes at that node, and as the sequence number changes, it generates the frog population to be initialised.



The frogs were divided into several memeplx and the pheromone levels were updated. If the paths are identified by ants, a message is sent to the origin node, and the best solution is determined by the fitness value reached. Otherwise, it sends the request to a randomly generated population of frogs. The fundamental idea is to calculate the fitness function over an initial population and then iteratively calculate the fitness of a new population, which is accomplished by performing operations on populations. Finally, we can compare all of the solutions to the best one. The best solution gives us the optimal values of the population, which we can use to find the best routing path.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 2 shows a detailed flowchart representation of the proposal. As a result, multiple paths to the source node are available. As a result, the shortest path is selected to deliver the packet to its destination. The proposed method is based on the SFLAO algorithm, in which packets are routed to the source node based on the best fitness value computation. In MANET, the best fitness value satisfying the QoS is used to select a path. Data is transmitted from source to destination in 48-byte packets. SFLAO algorithm calculates and compares QoS parameters such as throughput, energy consumption, routing overhead, end-to-end delay, and packet delivery ratio with the existing algorithm. The following are the overall steps for the SFLAO algorithm.

PROPOSED SFLAO algorithm

STEP 1: Begin;
STEP 2: Generate random population of P frogs;
STEP 3: For each individual i in P calculate fitness (i) ;

STEP 4: Sort the population P in descending order of their fitness;

STEP 5: Divide P into o m memeplexes;

STEP 6: For each memeplex:

a) Determine the fitness of local best(Xl**b**)frog as flb and the fitness of local worst(X**w**) frogs as fw

b) Try to improve the position of worst frog using

$$change-in-frog-position(D_d)=rand()-x(X_{lb}X_w) \quad (1)$$

Eq. (1) with respect to the local best frog. If fitness improves, update the position of the worst frog.

Improve the worst frog's position. The step and new position are computed for the frog
with worst performance in the submemeplex PB and PW correspond to the positionsby

step size **S** = min{int[rand(**PB** − **P**W)], **S**max} for a positive step,

= max{int[rand(**PB** − **P**W)],−**S**max} for a negative step,

Determine S worst frog,S_{max};

Initialise the number of particles and generate its value randomly.
Step 7: Initialise ACO parameters.
Step 8: Generate solutions from each ant's random walk.
Step 9: Update the pheromone intensities using Eq.(3), where $\rho \rightarrow$ Pheromone evaporation coefficient.

$$z_{xy} \leftarrow (1-\rho)T_{xy} + \sum_q \nabla^m_{z_{xy}} \quad (3)$$

$\nabla^m_{z_{xy}} \rightarrow$ Amount of pheromone deposited.

q $\rightarrow$ Ant that deposit the pheromone.

x is the index for the subsystem, and y refers to the components in a subsystem.

STEP 10:While (not met closing condition) do
create ant solutions by applying local search
find fitness value corresponding to each ant
find best solution and update pheromones trails using equation 2
End while
Multiple routes are created by ACO
Now generate random population of particles
Repeat
For all particles
STEP11;Evaluate fitness value for local and global solutions
Select best position
End For
STEP12: Combine shuffles the evolved memeplexes;

STEP13: Sort the population P in descending order of their fitness;

STEP14: Check the criterion is satisfied;

STEP15:End;

4. EXPERIEMENTAL RESULTS

The experiment is carried out on a computer equipped with Windows 7, an Intel core i-3 processor, a network adapter with a frequency of 2.16 GHz, and 2 GB of RAM. NS-2 is used to implement the SFLAO. This is a well-known and popular network simulator tool. This tool is used in MANET, wireless sensor networks, and other areas.

Parameter	Value
Simulator	NS-2
Number Of Nodes	10,20,30,40,50,60
Minimum Speed	1 m/s
Pause Time	10 s
Simulation Time	300 s
Transmission Range	250M
Node Movement Model	Random Way Point
Traffic Type	CBR(UDP)
Data Payload	512 Bytes/Packet

Table 1: Simulation Parameter

Packet delivery ratio is The ratio of the number of distinct packets received successfully at the sinks to the total number of packets generated at the source node; although a packet may reach the sinks multiple times, these redundant packets are treated as only one distinct packet. Typically, sending the package in these networks is done step by step or in several jumps. The packet delivery ratio in this type of network is calculated using formula 2.

$$PDR = \frac{\sum The\ number\ of\ received\ packets\ in\ destination}{\sum The\ number\ of\ sent\ packets}$$

PDR is a packet delivery ratio calculated by dividing the number of received packets by the number of sent packets.

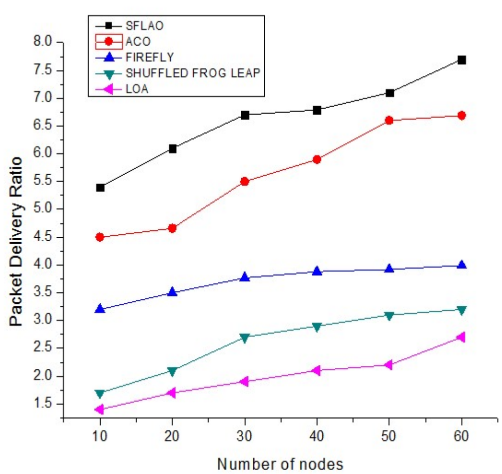


Figure 3: Packet Delivery Ratio

Figures 7 and 8 depict the simulation we ran on the proposed method of nodes at 10, 20, 30, 40, 50, and 60 seconds. The simulation results show that the proposed method SFLAO outperforms the FIREFLY,SFLA,LOA,ANTHOCNET algorithms. The times in terms of ratio packet delivery and this argument demonstrated the proposed model's superior performance.

Energy Consumption takes into account the total energy consumed in packet delivery, including transmitting, receiving, and idling energy consumption of all nodes in the network.

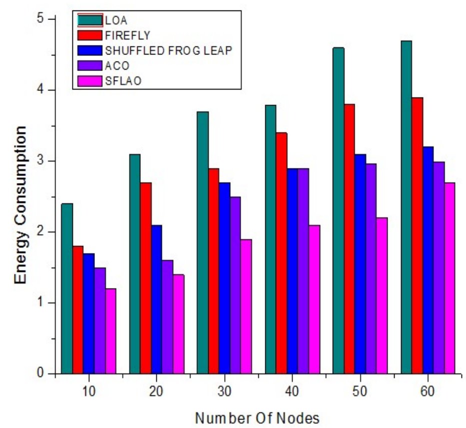


Figure 4: Energy Consumption

The average energy required for transmitting, receiving, or forwarding operations of a packet to a network node over a given time period is referred to as energy consumption. The proposed SFLAO outperformed the Anthocnet, LOA, Firefly, and SFLA power consumption algorithms. In x-axis number of nodes is taken and power consumption is taken as y-axis. The graph clearly shows that the proposed hybrid ACO consumes less energy than the existing method.

End To End Delay: The transmission delay of a network is defined as the ratio of the total number of available nodes to the sum of delays in each node. It is shown in the equation (27),

$$Delay = \frac{\sum (T^t - T^r)}{q} \qquad (27)$$

Where t T denotes the time taken to transmit packets, r T denotes the time taken to receive packets, and q denotes the total number of available nodes. End to end delay refers to the standard time required for packet transmission from source to destination nodes across the system. Figure 1 depicts the end-to-end delay of the MANET network when transferring data. The system avoids intermediate attacks while taking the shortest time

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

from start to finish. The proposed system uses a lower end-to-end value when moving data from source to destination when compared to the Qos process of algorithms Anthocnet, LOA, Firefly, and SFLAO Algorithms. When compared to other algorithms, SFLAO network nodes and calculates the value of the nearest node, resulting in very little delay. The delayed metric is depicted in Figure 5. As a result, the discussed SFLAO algorithm achieves less end-to-end delay..

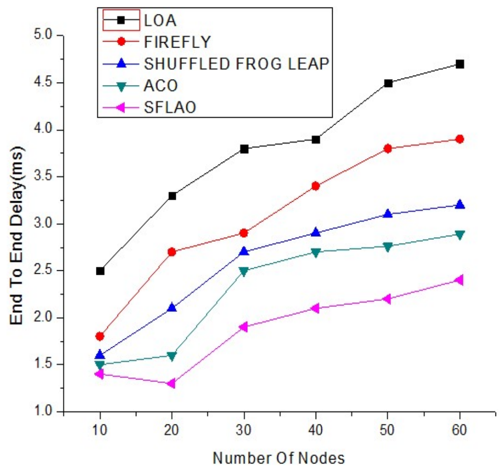


Figure 5:End to End Delay

Throughput: Defined As the number of bytes successfully received by the destination. T = / (%), where - Packets Received, - Packets Departed. Figure 9 depicts the variation in average throughput for algorithms. While node mobility increased as (10,20,30,40,50,60), the routing algorithm's throughput decreased. SFLAO outperforms the ACO, LOA, Firefly, and Shuffled Frog Leap algorithms in terms of throughput. As it chooses the most direct route to the destination. This route has less delay and more energy level than other routes; therefore the link is more stable which leads for fewer packets dropped. throughput of the network concerning packet size for Anthocnet, LOA, Firefly, SFLA Increase in throughput represents the growth in the data flow rate. The throughput is determined by the network statistics charge measured. It measures the amount of data transferred over a network in one unit of time. The SFLAO algorithm achieves the highest throughput for packet sizes of 250 bits. Thus, the analysis shows that SFLAO has higher throughput than Anthocnet, LOA, Firefly, and SFLAO algorithms.

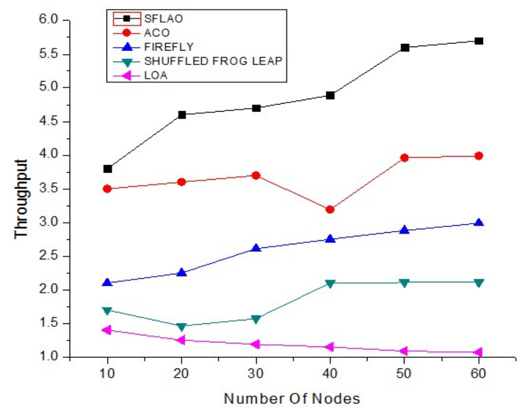


Figure 6: Throughput

Routing overhead : In any wireless network, the routing information from the sender to the destination is found by flooding the network with control packets, which is responsible for finding the destination requested by the sender. Control message overhead is depicted in Figure 8 for all approaches. As shown in the graph, SFLAO has lower control message overhead than Anthocnet, LOA, Firefly, and SFLAO. The LOA algorithm performs poorly in this case when compared to the other algorithms, which worsens the situation when the number of nodes increases from 30 to 40 due to optimization algorithms updating simulation parameters for different environments. As a result, data packets in the routing game are stochastically routed through various time slots. This process reduces routing overhead by employing the proposed strategy.

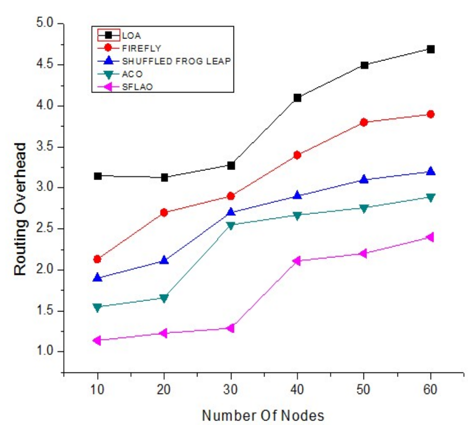


Figure 7: Routing Overhead

Network Lifetime : It goes without saying that the proposed routing algorithm improves network lifetime. The network's lifetime is calculated by counting the number of alive nodes at a given point in time. When compared to existing techniques, it is discovered that the network with the proposed routing algorithm has more alive nodes with increasing simulation time. As a result, the proposed SFLAO algorithm outperforms standard performance metrics, and the section that follows summarises the research findings. According to the analysis, the proposed approach consumes less energy than the comparative approaches. The network's energy consumption and lifetime are intertwined. When energy consumption is reduced, the network's lifetime can be extended. Energy consumption can be reduced by using various techniques, and this work uses optimal route selection to save energy. SFLAO's energy consumption is low in this location. The results are shown in the following figure after the network lifetime has been examined and the SFLAO lifetime network can be improved.

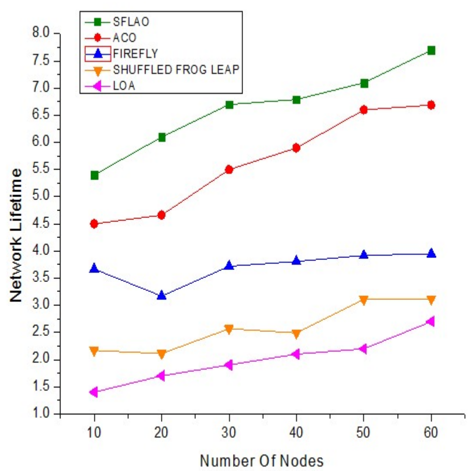


Figure 8: Network Lifetime

5. CONCLUSION

The difficulties in mobile ad hoc networks are in finding a path between communication end points that meets the user's QoS requirements while maintaining consistency. The algorithm includes both reactive and proactive elements. During a data session, the option of selecting multiple paths during the reactive path setup phase can be used to build the link between the source and destination. Techniques are more robust, and computation times are moderate to short. The algorithm's complexity is high when compared to other techniques, but even with high complexity, processing time is quick. Many of them fail to take into account the multiconstraint problems of QoS routing in MANET. Even though some of them deal with multi-constraint QoS parameters, they are scalability constrained. They also do not investigate service differentiation through service classes for QoS provisioning. In order to achieve an effective and efficient SFLAO in MANETs.

REFERENCES

[1] . K. A. Gupta, Harsh Sadawarti, K. A. Verma, “Performance analysis of AODV, DSR and TORA Routing Protocols,” *International Journal of Engineering and Technology (IJET),* ISSN: 1793-8236, Article No. 125, Vol.2 No. 2, April, pp. –Yaode 2010.

[2] . Suman Banik, Bibhash Roy, Biswajit Saha and Nabendu Chaki, **“**Design of QoS Routing Framework based on OLSR Protocol,” ARTCOM 2010, Kochin, Kottyam Kerala, IEEE Explorer, pp-171-73, 2010.

[3] . C.Perkins and P.Bhagwat, “Highly dynamic destination-sequenced distance-vector routing(dsdv) for mobile computer”, ACM Sigcomm94, August 1994. [5] Sun, W., Tang, M., Zhang, L., Huo, Z. & Shu, L., “A survey of using swarm intelligence algorithms in IoT”, Sensors, Vol. 20, No. 5, 2020, pp. 1420.

[4] . Ramanjyot Kaur, “Swarm Intelligence based Routing Algorithms for MANETs: A Review”, International Journal of Science Technology & Engineering, Vol. 2, No. 07, January 2016.

[5] . Alexandros Giagkos, Myra S. Wilson ,“BeeIP – A Swarm Intelligence based routing for wireless ad hoc networks”, Information Sciences, Vol. 265, No. 1 May 2014, pp. 23-35.

[6] . R. Bhaskar, J. Herranz, and F. Laguillaumie,“Efficient authentication for reactive routingprotocols”, in AINA06: Proceedings of the 20th International Conference on Advanced Information Networking and Applications - Vol. 2, No. 06, Washington, DC, USA: IEEE Computer Society, 2006, pp. 57– 61

[7] . Z. YaztiDemetrios, A Glance at Quality of Services in Mobile Ad-Hoc Networks, Technical Report. 2001.

[8] . E.M.Royer and C.K. Toh, “A review of current routing protocols for ad hoc mobile wireless networks” IEEE Personal Communications, 6(2): 46-55 , 1999

[9] . Mccomb, K, et al. Female lions can identify potentially infanticidal males from their roars. Proc. R. Soc. Lond. Ser B: Biol. Sci. 1993;252 (1333)59–64.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[10] . Schaller GB. The Serengeti lion: a study of predator–prey relations. Wildlife behavior and ecology series. Chicago, Illinois, USA: University of Chicago Press; 1972.
[11] . J. B. Buck, “Synchronous rhythmic flashing of fireflies,” The Quarterly Review of Biology, vol. 13, no. 3, pp. 301–314, 1938.
[12] . A. Tyrrell, G. Auer, and C. Bettstetter, “Fireflies as role models for synchronization in ad hoc networks,” in Proceedings of the 1st International Conference on Bio Inspired Models of Gianni Di Caro*, Frederick Ducatelle and Luca Maria Gambardella,”Network, Information and Computing Systems”, 2006, pp. 1–7.
[13] . Gianni Di Caro, Frederick Ducatelle and Luca Maria Gambardella,”AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks”, European Transactions On Telecommunications,Euro. Trans. Telecomms. 2005; 16:443–455
[14] . Shahab Kamali1 and Jaroslav Opatrny,”A Hybrid Ant-Colony Routing Algorithm for Mobile Ad-Hoc Networks”, Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2009, pp. 1337–1354, 2009.
[15] . Tong, X., Y. Ji, J. Lin, J. Zhu, F. Sun, Y. Zhong, Y. Yang & X. Zhu, Cooperative spectrum sensing based on a modified shuffled frog leaping algorithm in 5G network. Physical Communication, 2017. 25: pp. 438-444.
[16] . Sharma, T.K. & M. Pant, Opposition-Based Learning Embedded Shuffled Frog-Leaping Algorithm, in Soft Computing: Theories and Applications. 2018, Springer. pp. 853-861.